BIAX Corporation v. Intel
Civil Action No. 2:05-cv-184-TJW

EXHIBIT 5
(PART 1)

US005765037A

United States Patent [19]

Morrison et al.

[11] Patent Number: 5,765,037

[45] Date of Patent: Jun. 9, 1998

[54] **SYSTEM FOR EXECUTING INSTRUCTIONS WITH DELAYED FIRING TIMES**

[75] Inventors: **Gordon Edward Morrison**, Denver; **Christopher Bancroft Brooks**; **Frederick George Gluck**, both of Boulder, all of Colo.

[73] Assignee: **Biax Corporation**, Palm Beach Gardens, Fla.

[21] Appl. No.: **480,841**

[22] Filed: **Jun. 7, 1995**

Related U.S. Application Data

[62] Division of Ser. No. 254,687, Jun. 6, 1994, Pat. No. 5,517,628, which is a division of Ser. No. 93,794, Jul. 19, 1993, abandoned, which is a continuation of Ser. No. 913,736, Jul. 14, 1992, abandoned, which is a continuation of Ser. No. 560,093, Jul. 30, 1990, abandoned, which is a division of Ser. No. 372,247, Jun. 26, 1989, Pat. No. 5,021,945, which is a division of Ser. No. 794,221, Oct. 31, 1985, Pat. No. 4,847,755.

[51] **Int. Cl.**[6] **G06F 9/40**

[52] **U.S. Cl.** **395/557**; 395/391; 395/559; 395/800.21; 395/580

[58] **Field of Search** 395/375, 550, 395/391, 595, 557, 559, 800.21, 580

[56] **References Cited**

U.S. PATENT DOCUMENTS

4,200,912	4/1980	Harrington et al.	395/742
4,229,790	10/1980	Gilliland et al.	395/671
4,247,894	1/1981	Beismann et al.	395/737
4,250,546	2/1981	Boney et al.	395/735
4,342,078	7/1982	Tredennick et al.	395/387
4,430,707	2/1984	Kim	395/670
4,466,061	8/1984	DeSantis	395/676
4,532,589	7/1985	Shintani et al.	395/393
4,598,400	7/1986	Hillis	370/400
4,833,599	5/1989	Colwell et al.	395/583

OTHER PUBLICATIONS

Chang et al., "801 Storage: Architecture and Programming," IBM T.J. Watson Research Center, 1988, 0734-2071/88/0200-0028, pp. 29-50.

Colwell et al., "A VLIW Architecture for a Trace Scheduling Compiler." Second International Conference on ASPLOS-II, Oct. 5-8, 1987, pp. 1-23.

Davis, "Tomorrow's Computers—The Challenges," IEEE Spectrum, Nov. 1981?, pp. 94-99.

Ellis, "Bulldog: A Compiler for VLIW Architectures," MIT, 1986.

Gross et al., "Optimizing Delayed Branches," IEEE, 15th Annual Workshop on Microprogramming, 1982, 0194-1895/82/0000/0014, pp. 114-120.

Hagiwara et al., "A Dynamically Microprogrammable Computer with Low-Level Parallelism," IEEE, Jul. 1980, 0018-9340/80/0700-0577, vol. C-29, No. 7, pp. 577-594.

Hennessy et al., "The MIPS Machine," IEEE, 1982, CH1739-2/82/0000-0002, vol. C-24, pp. 2-7.

Hennessy et al., "Postpass Code Optimization of Pipeline Constraints," ACM Transactions on Programming Languages and Systems, Jul. 1983, ACM 0164-0925/83/0700-0422, vol. 5, No. 3, pp. 442-448.

(List continued on next page.)

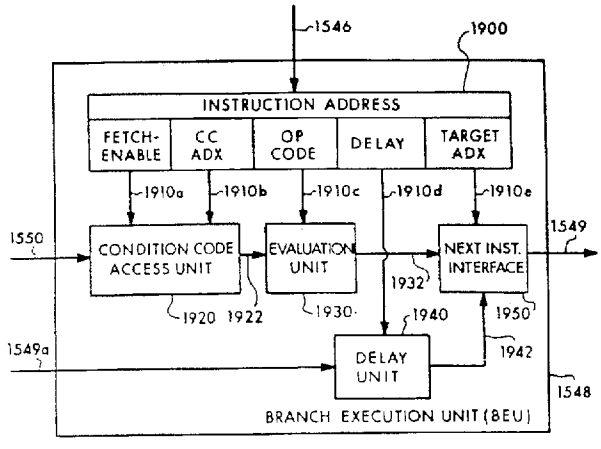
Primary Examiner—Kenneth S. Kim

Attorney, Agent, or Firm—Fish & Richardson P.C.

[57] **ABSTRACT**

A system and method reorder instructions for effecting faster branch execution. A processor element is coupled to receive stored instructions in a first order, and to process the received instructions in a different order, the processing occurring after each stored instruction of a first type is issued, and after a delay time, after each stored instruction of a second type is issued. The delay time is based on a delay value associated with the second type of instructions. In particular, the instructions include branch and non-branch instruction wherein firing time information identifies a time of execution of the branch instruction which is a variable number of instructions cycles prior to a time of execution of a last to be executed instruction in a basic block. Accordingly, branch instructions can be completely executed no later than during the processing of the last to be executed non-branch instruction in the basic block thereby speeding up overall processing of the software program by the system.

29 Claims, 17 Drawing Sheets



OTHER PUBLICATIONS

Hennessy, "VLSI Processor Architecture," IEEE Transactions on Computers, 0018–9340/84/1200–1221, vol. C–33, No. 12, pp. 1221–1246, 1984.

Hennessy, "VLSI RISC Processors," *VLSI Systems Design*, Oct. 1985, pp. 22–24, 28, 32.

McDowell et al., "Processor Scheduling for Linearly Connected Parallel Processors," IEEE Transactions on Computers, 1986, 0018–9340/86/0700–0632, vol. C–35, No. 7, pp. 633–638.

McDowell, "A Simple Architecture for Low Level Parallelism," IEEE, 1983, 0190–3918/83/0000/0472, pp. 472–477.

McDowell, "SIMAC: A Simple Multiple ALU Computer," University of California, San Diego, 1983.

Patterson et al., "The Case for the Reduced Instruction Set Computer," University of California, Berkeley, pp. 25–32.

Patterson , "Reduced Instruction Set Computers," Communications of the ACM, Jan. 1985, vol. 28, No. 1, pp. 8–21.

Patterson, "Microprogramming," pp. 50–57.

Radin, "The 801 Minicomputer," *Computer Architecture News*, Apr. 1982, ACM 0–89791–066–4 82/03/0039, vol. 10, No. 2, 39–47.

Tomita et al., "A User–Microprogrammable, Local Host Computer with Low–Level Parallelism," Communications of the ACM, 0149–7111/83/0600/0151, pp. 151–157, 1983.

Alpha Architecture Reference Manual, Digital, 1992.

PowerPC 601 RISC Microprocessor User's Manual, IBM, 1993.

Intel MCS–80 User's Manual, Intel, Oct. 1977.

Heinrich, *MIPS Microprocessor R4000 User's Manual*, MIPS Technologies, Inc., 1993.

MC68030 Enhanced 32–Bit Microprocessor User's Manual, Second Edition, Motorola, Inc., 1989.

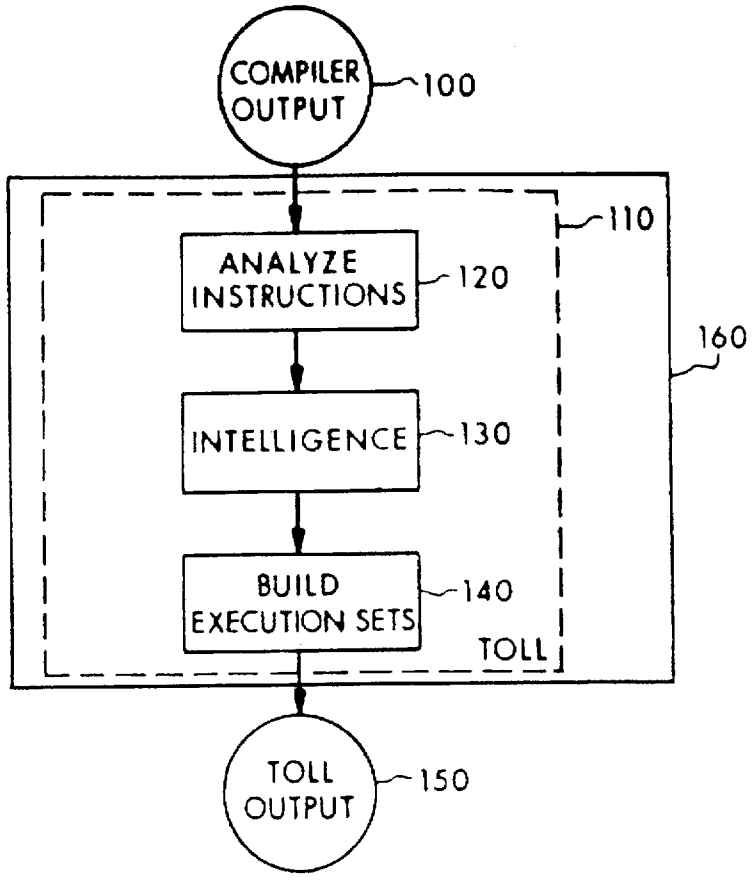


FIG. 1

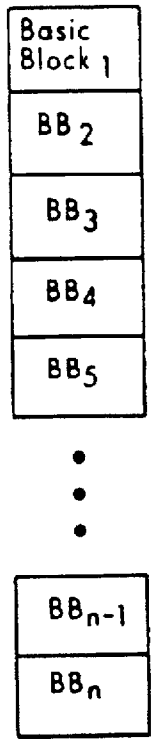


FIG. 2
PRIOR ART

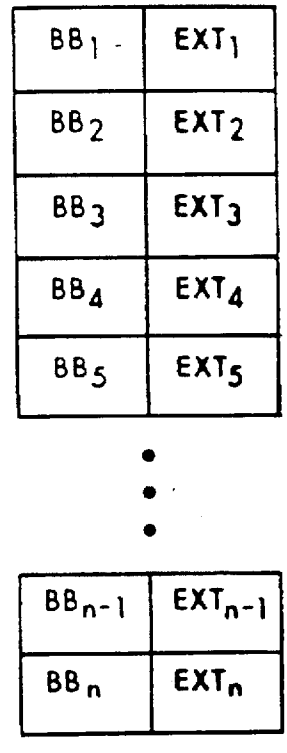


FIG. 3

FIG. 4

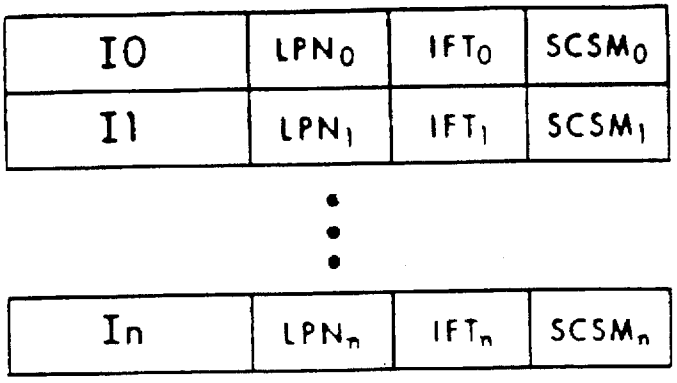
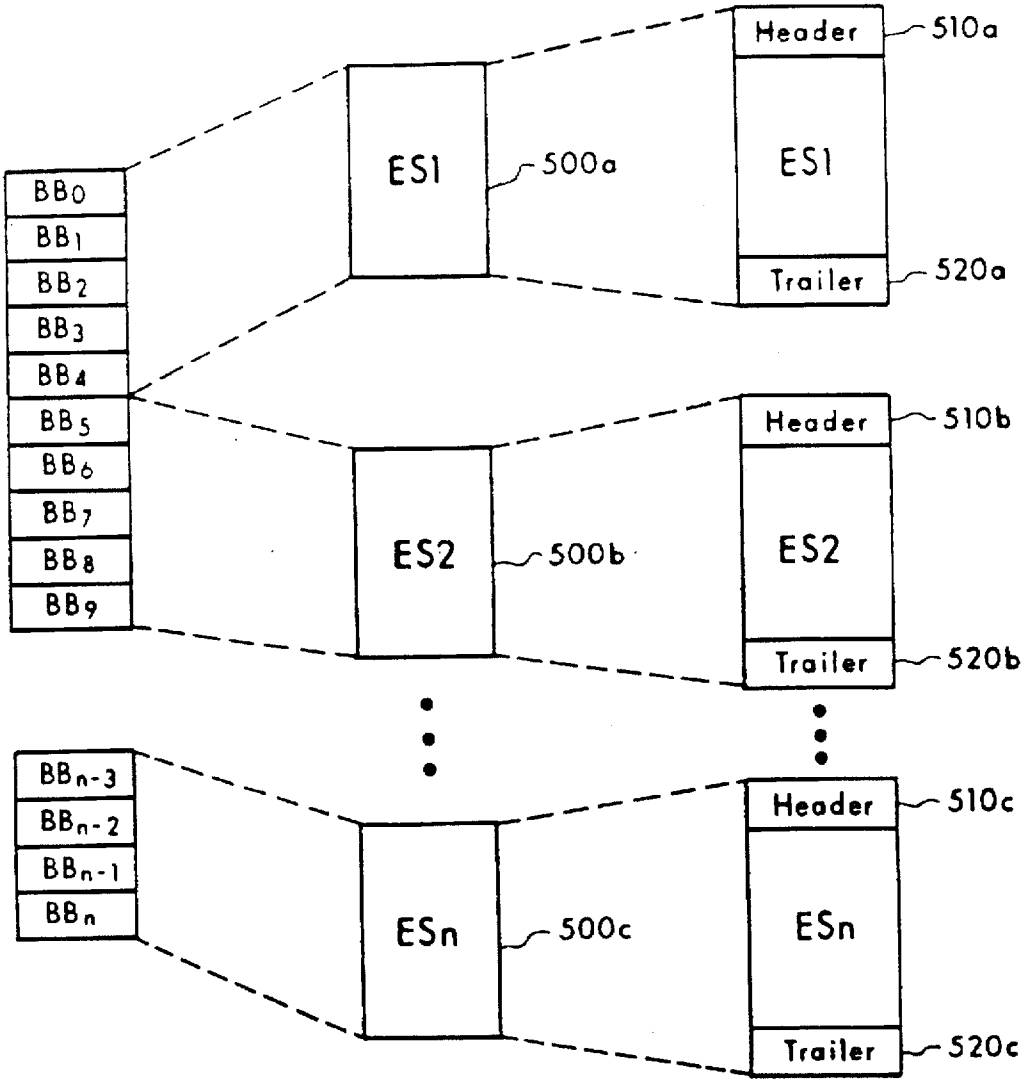

I0
LPN0
IFT0
SCSM0
I1
LPN1
IFT1
SCSM1
In
LPNn
IFTn
SCSMn

FIG. 5


BB0
BB1
BB2
BB3
BB4
BB5
BB6
BB7
BB8
BB9
BBn-3
BBn-2
BBn-1
BBn
ES1
500a
ES2
500b
ESn
500c
Header
510a
ES1
Trailer
520a
Header
510b
ES2
Trailer
520b
Header
510c
ESn
Trailer
520c

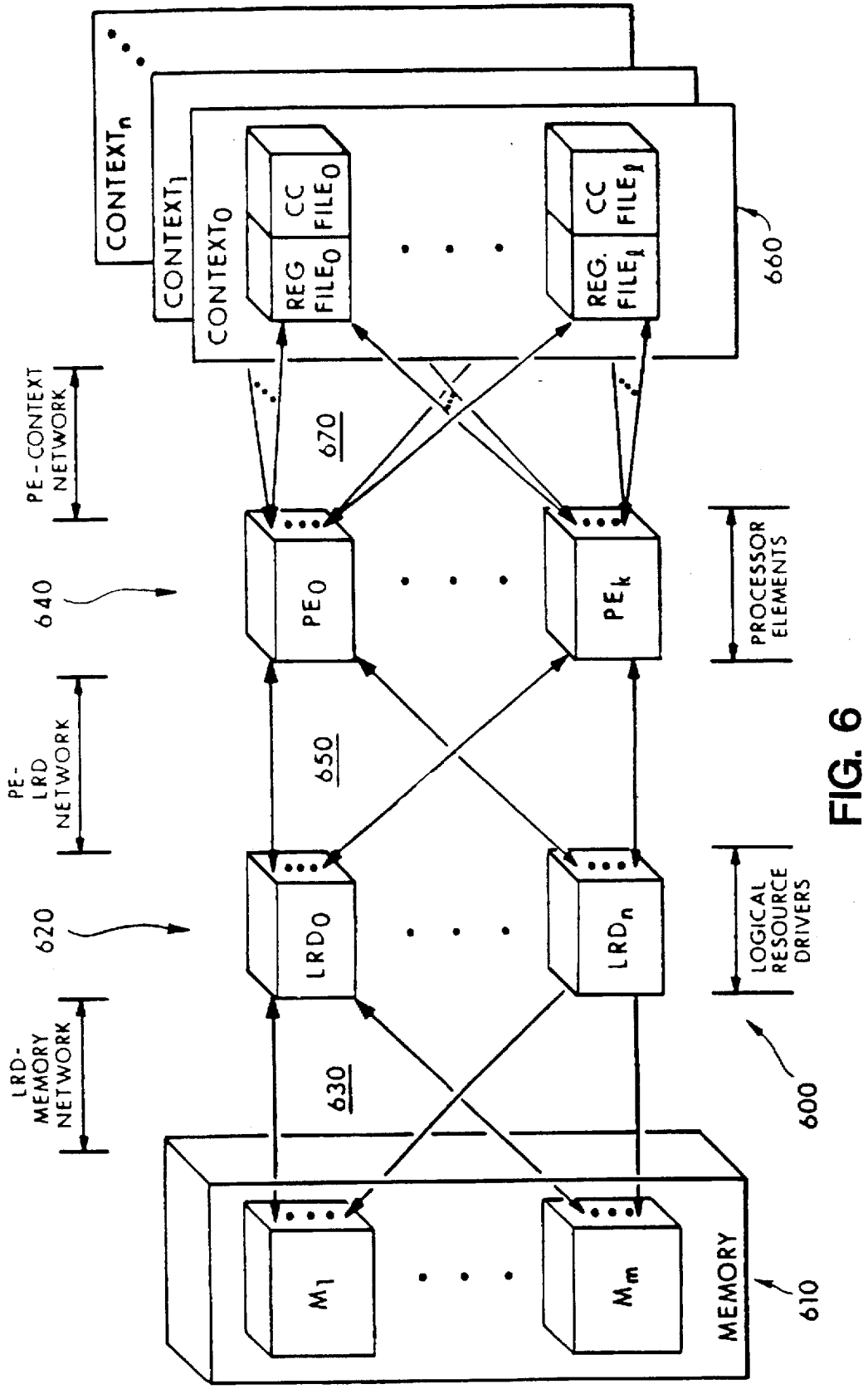

CONTEXT$_n$
CONTEXT$_1$
CONTEXT$_0$
REG FILE$_0$
CC FILE$_0$
REG. FILE$_j$
CC FILE$_j$
660
670
PE-CONTEXT NETWORK
640
PE$_0$
PE$_k$
PROCESSOR ELEMENTS
650
PE-LRD NETWORK
620
LRD$_0$
LRD$_n$
LOGICAL RESOURCE DRIVERS
630
LRD-MEMORY NETWORK
600
M_1
M_m
MEMORY
610

FIG. 6

FIG. 7a

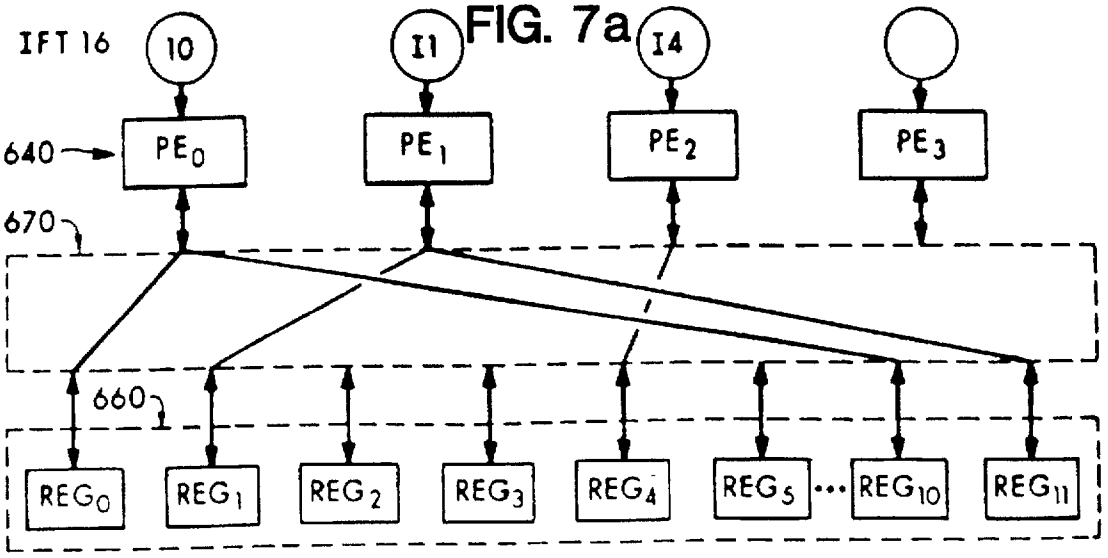


FIG. 7b

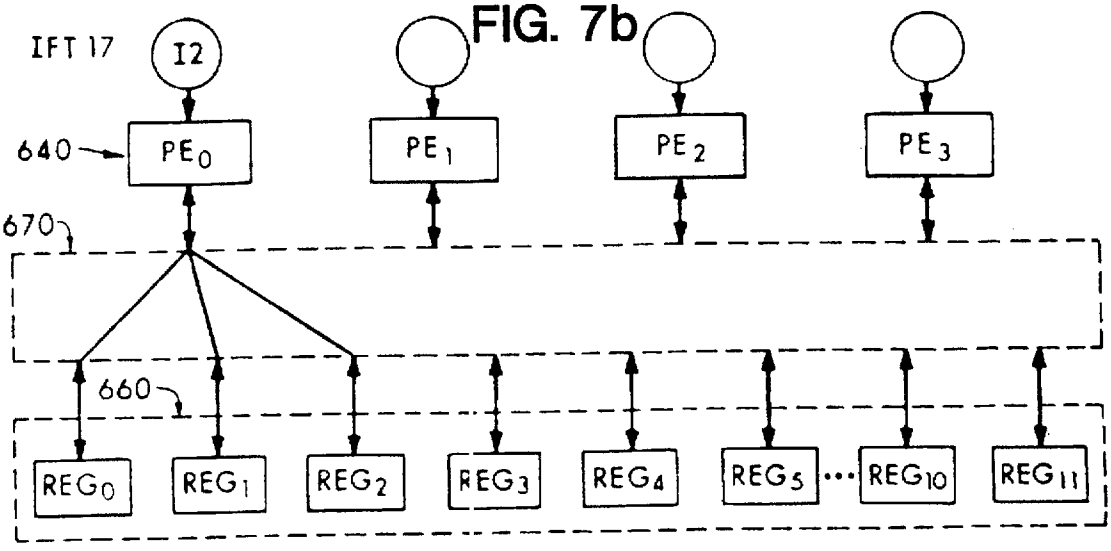
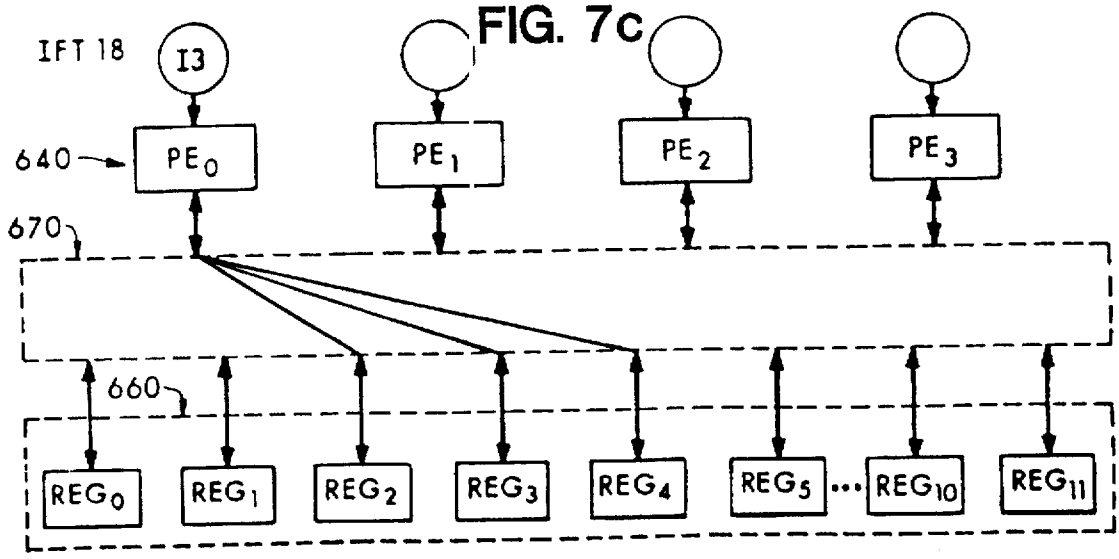


FIG. 7c



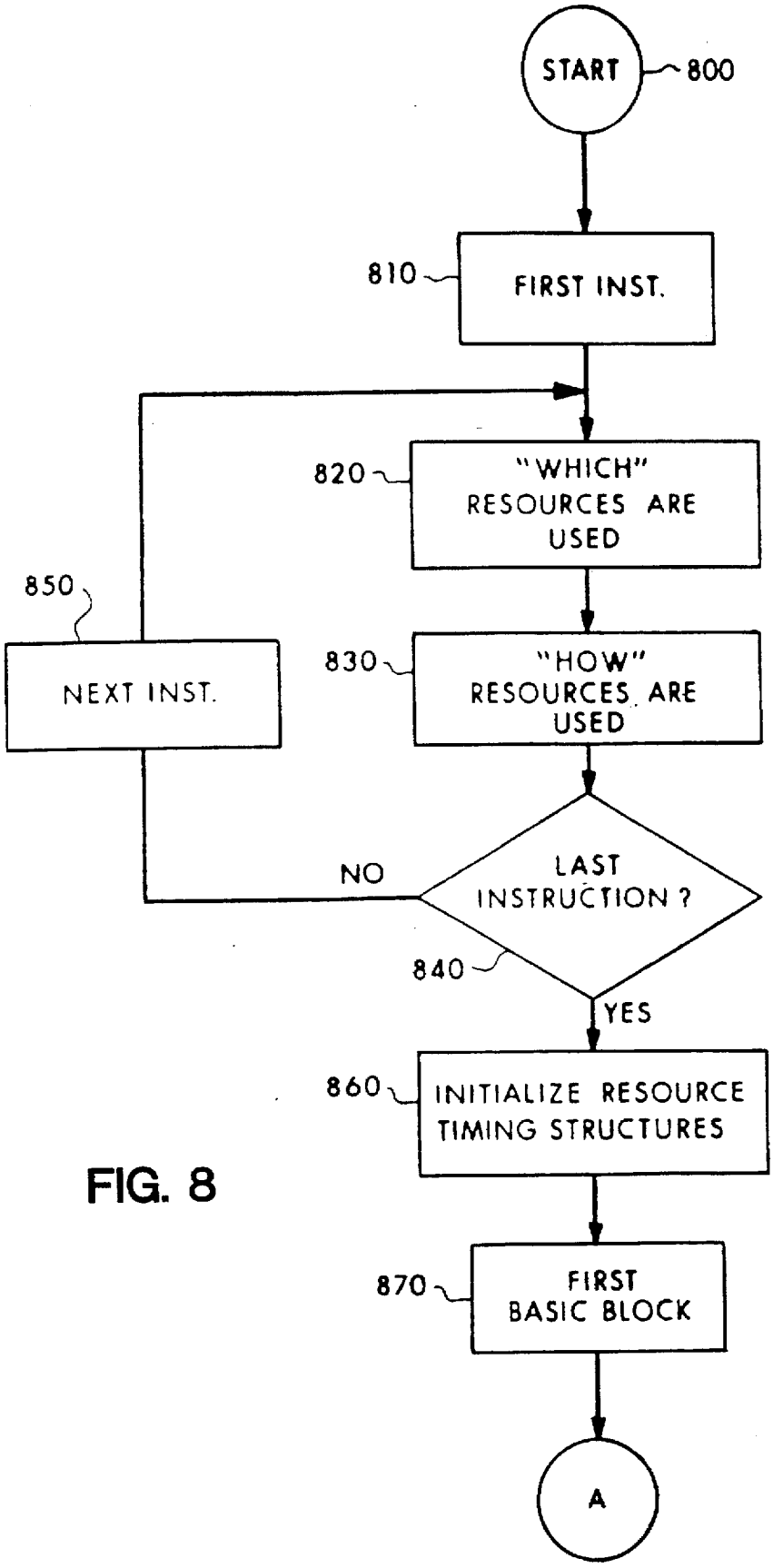

START
800
810
FIRST INST.
820
"WHICH" RESOURCES ARE USED
830
"HOW" RESOURCES ARE USED
840
LAST INSTRUCTION ?
NO
850
NEXT INST.
YES
860
INITIALIZE RESOURCE TIMING STRUCTURES
870
FIRST BASIC BLOCK
A

FIG. 8

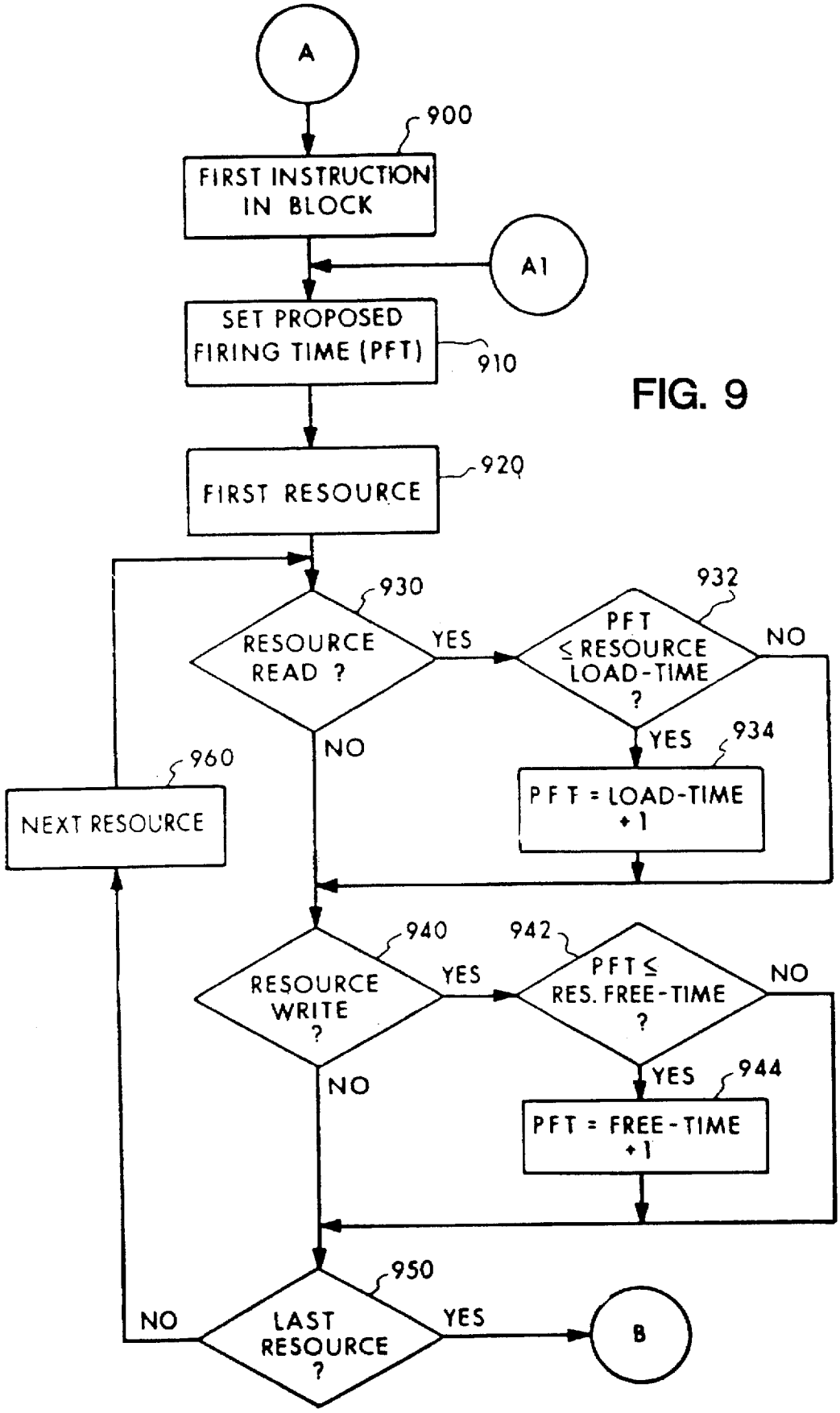

A
900
FIRST INSTRUCTION IN BLOCK
A1
SET PROPOSED FIRING TIME (PFT)
910
FIRST RESOURCE
920
930
RESOURCE READ ?
YES
932
PFT ≤ RESOURCE LOAD-TIME ?
NO
YES
934
PFT = LOAD-TIME + 1
NO
960
NEXT RESOURCE
940
RESOURCE WRITE ?
YES
942
PFT ≤ RES. FREE-TIME ?
NO
YES
944
PFT = FREE-TIME + 1
NO
950
LAST RESOURCE ?
NO
YES
B

FIG. 9

FIG. 10

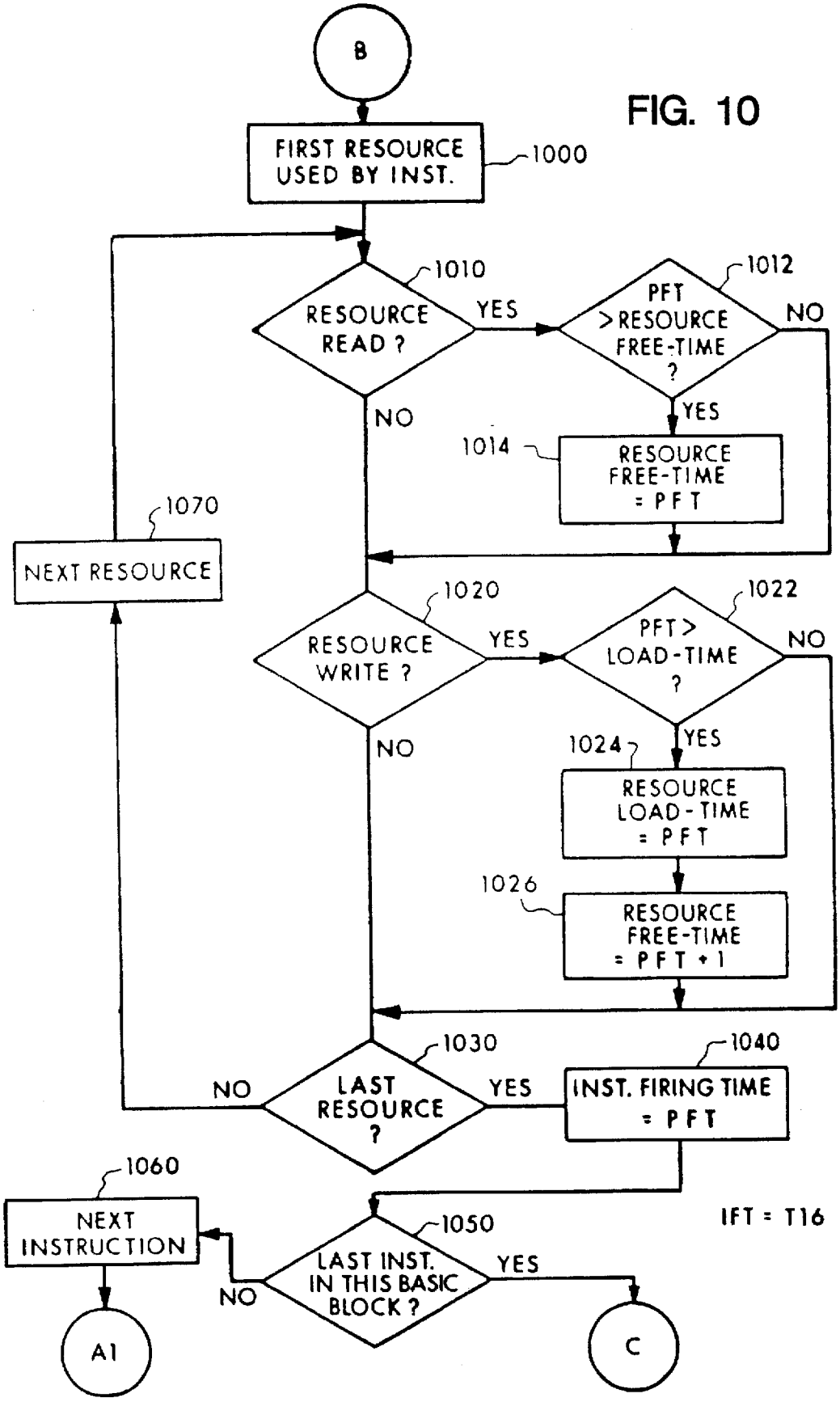
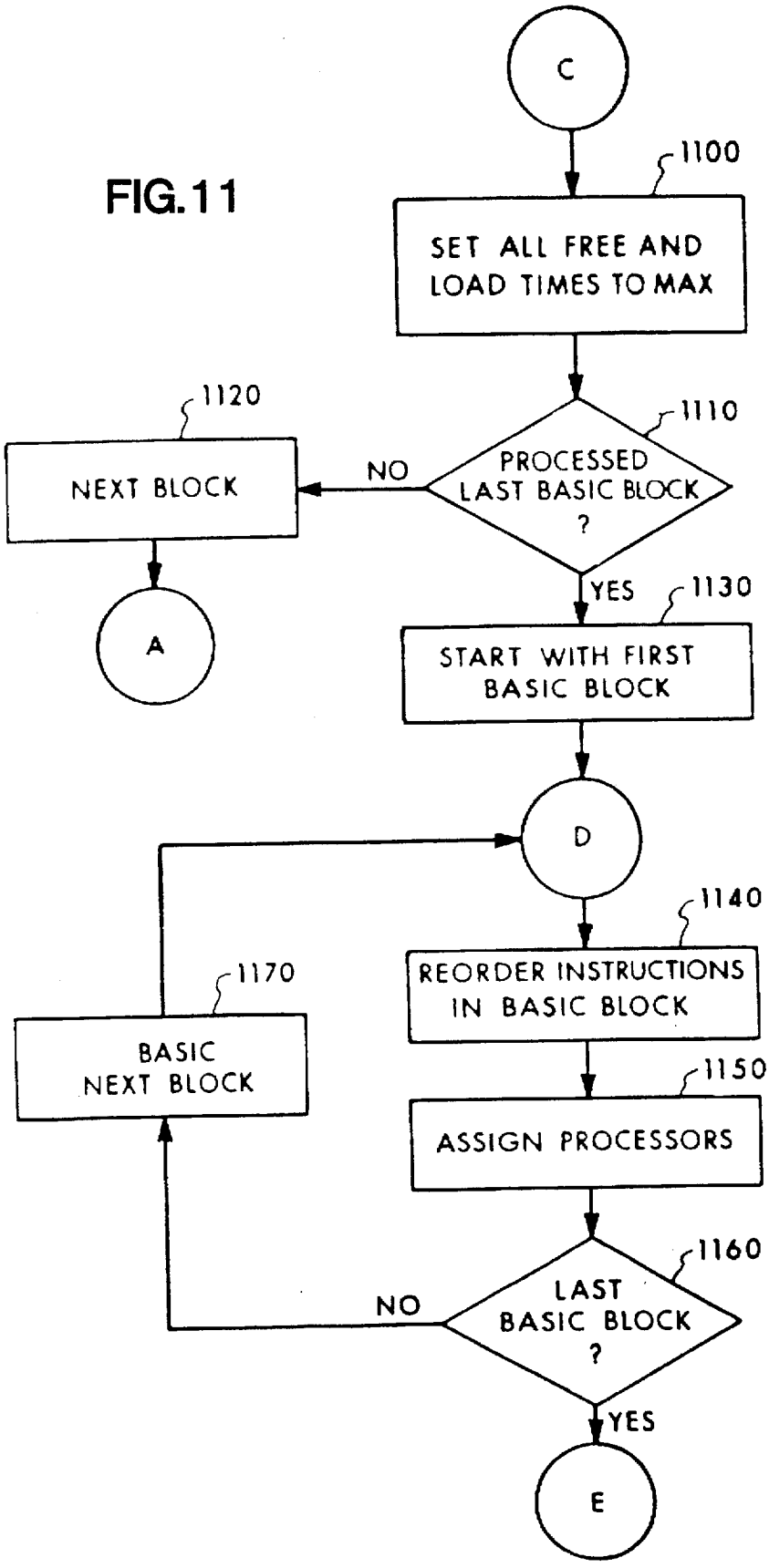

B
FIRST RESOURCE USED BY INST.
1000
1010
RESOURCE READ ?
YES
1012
PFT > RESOURCE FREE-TIME ?
NO
YES
NO
1014
RESOURCE FREE-TIME = PFT
1070
NEXT RESOURCE
1020
RESOURCE WRITE ?
YES
1022
PFT > LOAD-TIME ?
NO
NO
YES
1024
RESOURCE LOAD-TIME = PFT
1026
RESOURCE FREE-TIME = PFT + 1
1030
LAST RESOURCE ?
NO
YES
1040
INST. FIRING TIME = PFT
IFT = T16
1060
NEXT INSTRUCTION
1050
LAST INST. IN THIS BASIC BLOCK ?
NO
YES
A1
C

FIG. 11


C
1100
SET ALL FREE AND LOAD TIMES TO MAX
1110
PROCESSED LAST BASIC BLOCK ?
NO
1120
NEXT BLOCK
A
YES
1130
START WITH FIRST BASIC BLOCK
D
1140
REORDER INSTRUCTIONS IN BASIC BLOCK
1150
ASSIGN PROCESSORS
1160
LAST BASIC BLOCK ?
NO
1170
BASIC NEXT BLOCK
YES
E

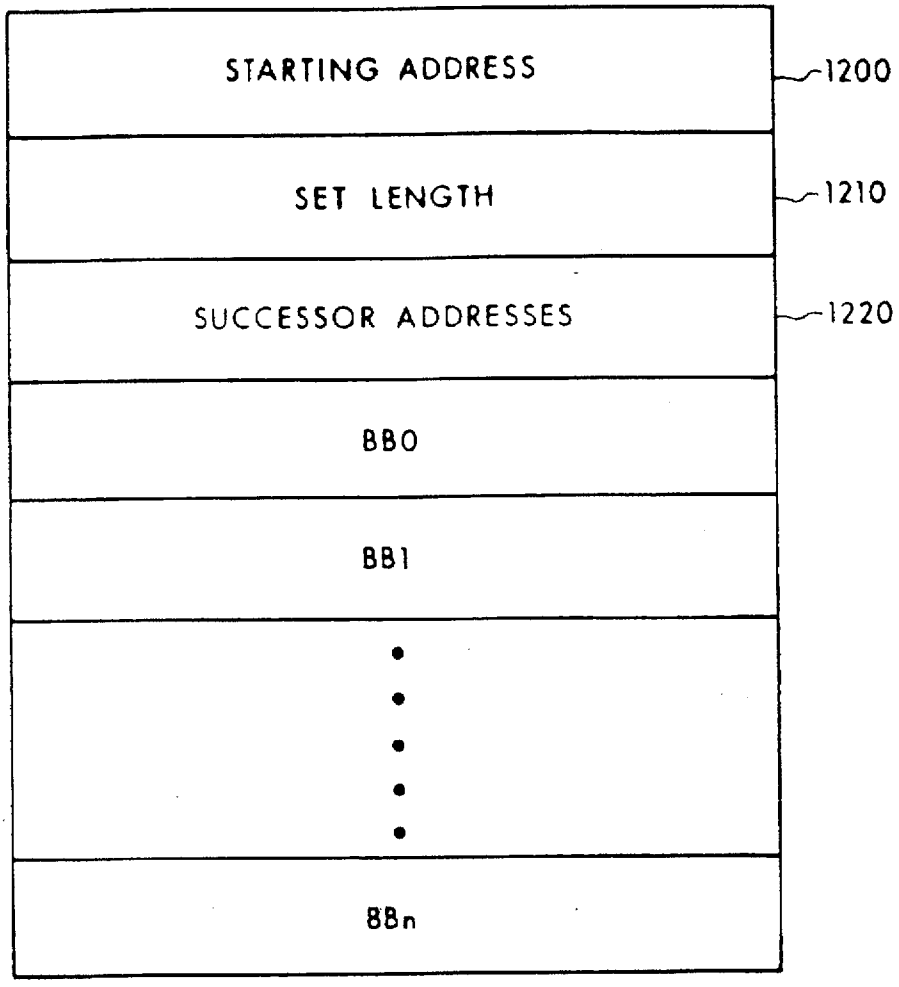

STARTING ADDRESS
1200
SET LENGTH
1210
SUCCESSOR ADDRESSES
1220
BB0
BB1
BBn

FIG. 12

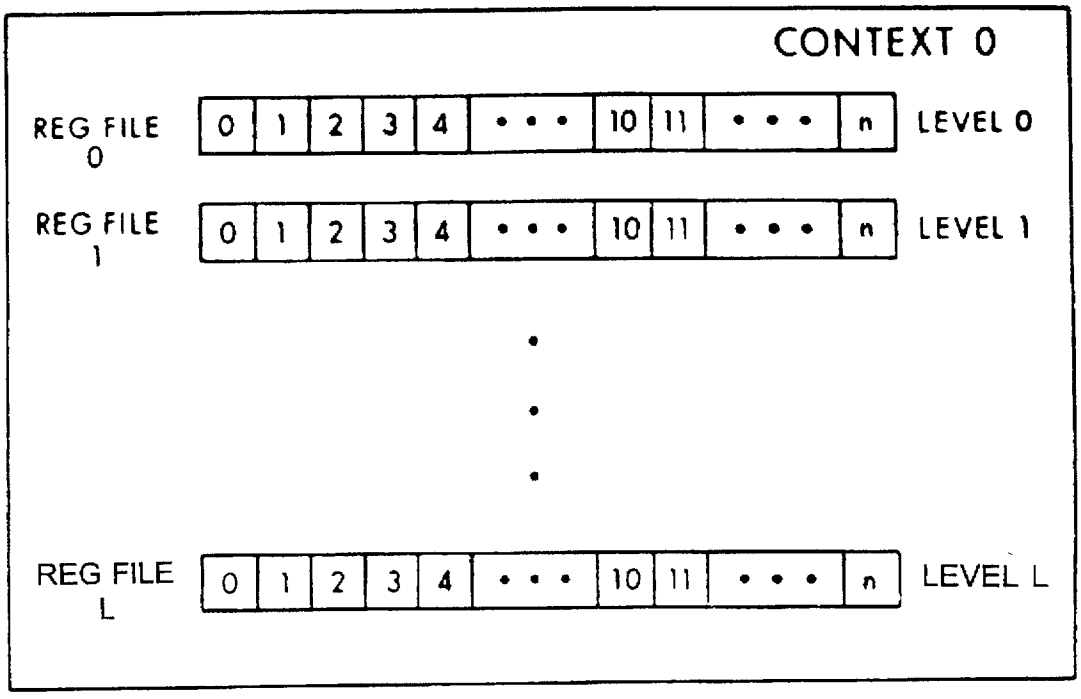

CONTEXT 0
REG FILE 0
REG FILE 1
REG FILE L
0 1 2 3 4 • • • 10 11 • • • n
LEVEL 0
LEVEL 1
LEVEL L

FIG. 13

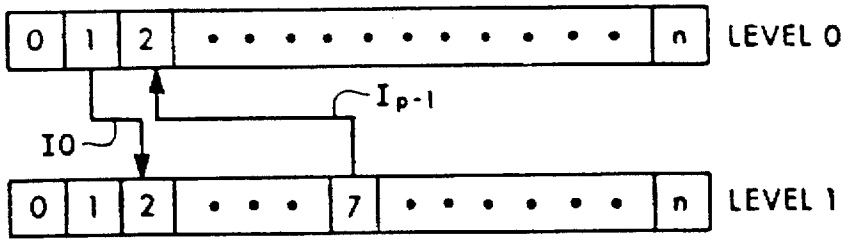

0 1 2 • • • • • • • • • • n
LEVEL 0
Ip-1
IO
0 1 2 • • • 7 • • • • • • n
LEVEL 1

FIG. 14

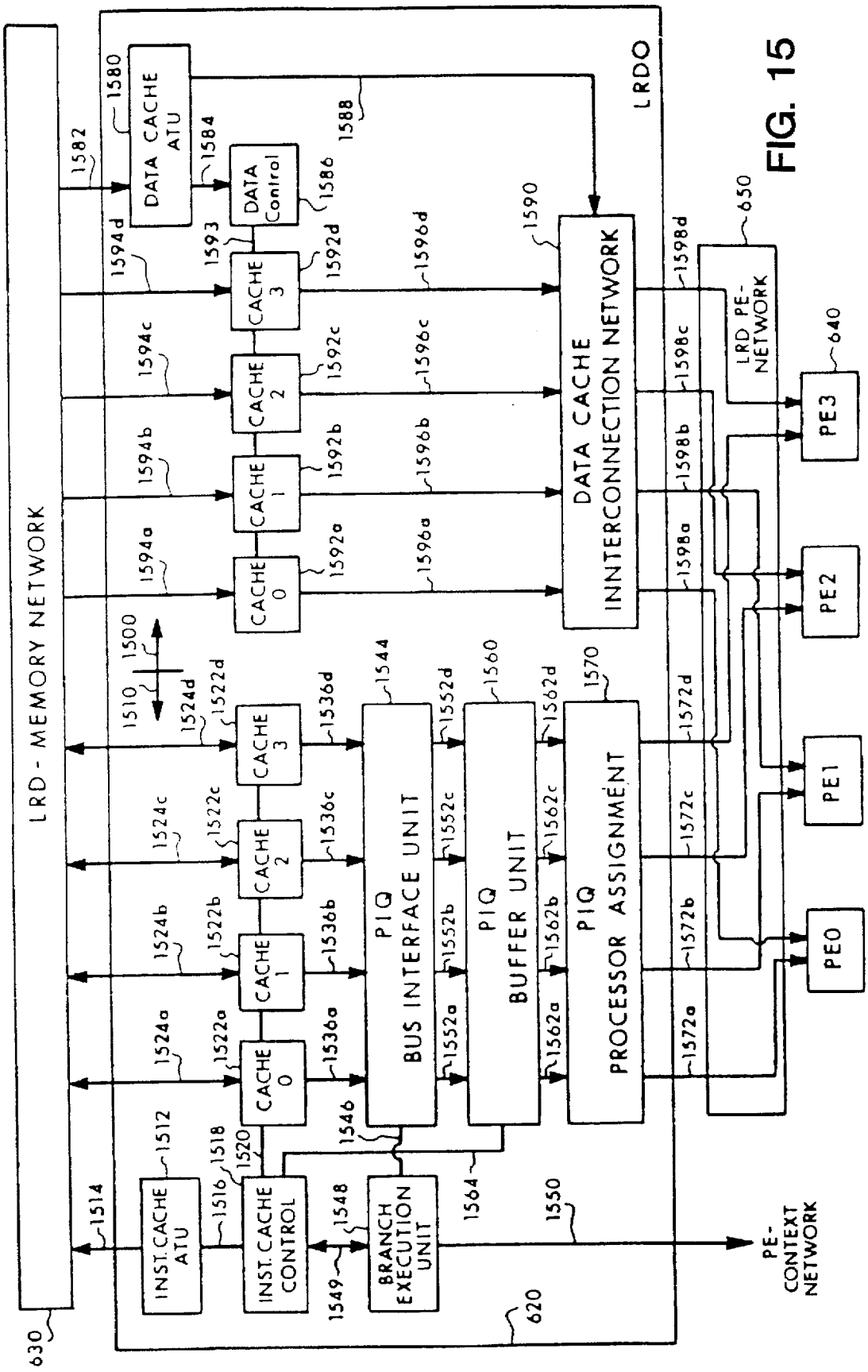

LRD - MEMORY NETWORK
LRDO
DATA CACHE ATU
DATA Control
CACHE 3
CACHE 2
CACHE 1
CACHE 0
DATA CACHE INNTERCONNECTION NETWORK
LRD PE-NETWORK
PE3
PE2
PE1
PE0
PIQ BUS INTERFACE UNIT
PIQ BUFFER UNIT
PIQ PROCESSOR ASSIGNMENT
INST. CACHE ATU
INST. CACHE CONTROL
BRANCH EXECUTION UNIT
PE-CONTEXT NETWORK
630
620
650
640
1580
1582
1584
1586
1588
1590
1593
1592d
1596d
1594d
1598d
1592c
1596c
1594c
1598c
1592b
1596b
1594b
1598b
1592a
1596a
1594a
1598a
1500
1510
1524d
1522d
1536d
1544
1552d
1560
1562d
1570
1572d
1524c
1522c
1536c
1552c
1562c
1572c
1524b
1522b
1536b
1552b
1562b
1572b
1524a
1522a
1536a
1552a
1562a
1572a
1512
1514
1516
1518
1520
1546
1548
1549
1564
1550

FIG. 15

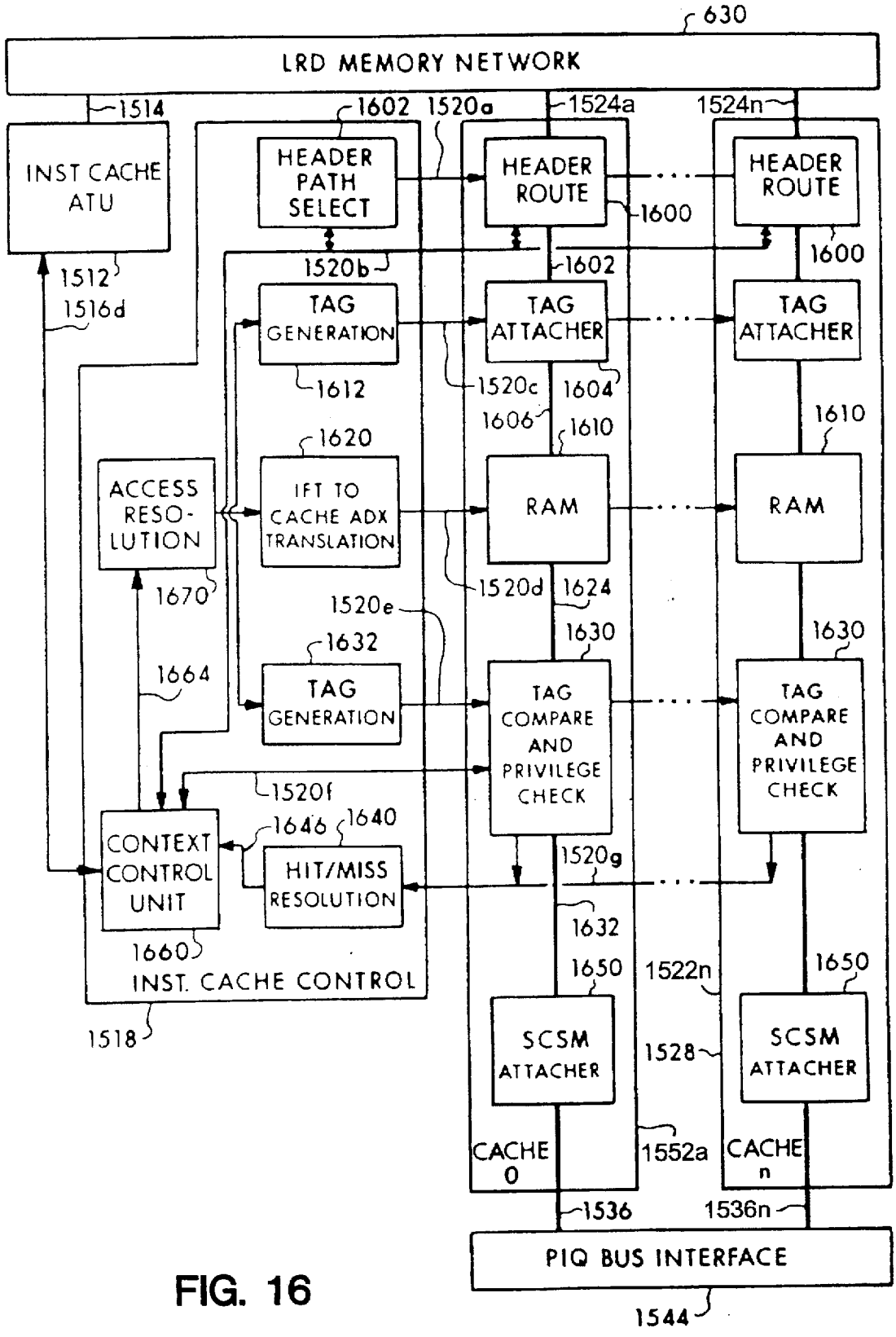

630
LRD MEMORY NETWORK
1514
INST CACHE ATU
1512
1516d
1602
HEADER PATH SELECT
1520a
1524a
HEADER ROUTE
1524n
HEADER ROUTE
1600
1600
1520b
1602
TAG GENERATION
1612
TAG ATTACHER
TAG ATTACHER
1520c
1604
1606
1610
1610
1620
ACCESS RESO-LUTION
IFT TO CACHE ADX TRANSLATION
RAM
RAM
1670
1520d
1624
1520e
1630
1630
1632
1664
TAG GENERATION
TAG COMPARE AND PRIVILEGE CHECK
TAG COMPARE AND PRIVILEGE CHECK
1520f
1646
1640
CONTEXT CONTROL UNIT
HIT/MISS RESOLUTION
1520g
1632
1660
INST. CACHE CONTROL
1650
1522n
1650
1518
SCSM ATTACHER
1528
SCSM ATTACHER
CACHE 0
1552a
CACHE n
1536
1536n
PIQ BUS INTERFACE
1544

FIG. 16

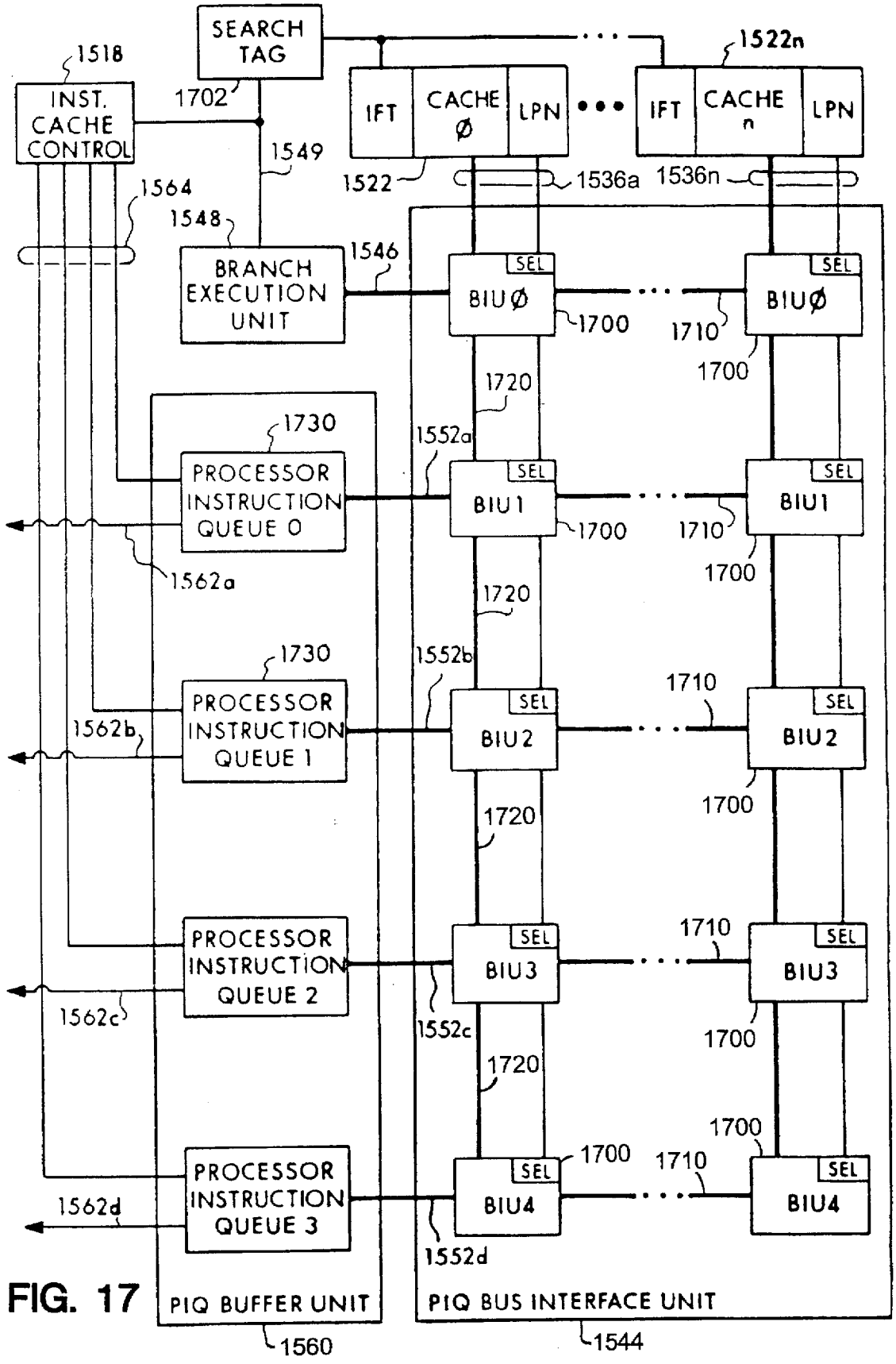

SEARCH TAG
1518
INST. CACHE CONTROL
1702
1549
1564
1548
BRANCH EXECUTION UNIT
1546
IFT
CACHE Ø
LPN
1522
1522n
IFT
CACHE n
LPN
1536a
1536n
SEL
BIUØ
1700
1710
1720
1730
PROCESSOR INSTRUCTION QUEUE 0
1562a
1552a
BIU1
PROCESSOR INSTRUCTION QUEUE 1
1562b
1552b
BIU2
PROCESSOR INSTRUCTION QUEUE 2
1562c
1552c
BIU3
PROCESSOR INSTRUCTION QUEUE 3
1562d
1552d
BIU4
PIQ BUFFER UNIT
1560
PIQ BUS INTERFACE UNIT
1544

FIG. 17